IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	RAJU, et al.	Examiner:	DENG, ANNA CHEN
Serial No.:	10/500919	Group Art Unit:	2191
Filed:	July 8, 2004	Docket No.:	20160.0001USWO
Title:	A UNIQUE VERSATILE EXECUTOR ENGINE WHICH CAN INTERPRET AND EXECUTE TRANSACTION STRUCTURES AND INFORMATION VIEWS TO BUILD INFORMATION SYSTEMS		

CERTIFICATE OF TRANSMISSION
I hereby certify that the papers listed below are being transmitted by EFS Web to: Mail Stop: Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on May 26, 2009.

By: Amy Doyle
Name: Amy Doyle

APPELLANT'S BRIEF ON APPEAL

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

52835
Customer Number

Dear Commissioner:

This Brief is presented in support of the Notice of Appeal filed on March 30, 2009 from the final rejection of claims 10-17 of the above-identified application, as set forth in the Office Action mailed September 29, 2008.

Please charge the fee of $540.00 for filing this brief to Deposit Account 50-3478. Please charge any additional fees or credit overpayment to Deposit Account No. 50-3478.

I. REAL PARTY IN INTEREST

The Real Party in Interest is Agile Labs Pvt.Ltd., of Bangalore, India.

II. RELATED APPEALS AND INTERFERENCES

The Assignee, the Assignee's legal representatives, and the Appellant are unaware of any other appeals or interferences that will affect, be directly affected by or have a bearing on the Board's decision in this Appeal.

III. STATUS OF CLAIMS

Claims 10-17 are pending. Claims 10-17 are rejected and claims 1-9 are cancelled. Claims 10-17 are the subject of this Appeal. Appendix VIII attached herewith provides a copy of claims 10-17 to be reviewed in this Appeal.

IV. STATUS OF AMENDMENTS

An Amendment under 37 C.F.R. §1.116 was filed on January 29, 2009. The Amendment was entered and considered, but was not found persuasive per the Advisory Action mailed on February 9, 2009.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Appellants have developed a tool that can interpret and execute transaction structures and information views to build information systems. The tool is capable of building an information system without writing a software program and is also able to implement new options and change existing options of the information system without any downtime (See Abstract).

All paragraphs referenced below are in reference to the originally filed specification, as amended by amendment filed on June 11, 2008.

Claim 10:

With regard to claim 10, a computer-readable medium having stored thereon a tool for building an information system (100) is provided (see paragraphs [0001] and [0048]). The tool (100) includes a builder component (130) that receives one or more transaction structures and one or more information views that form a business process, and creates a plurality of definitions

using the one or more transaction structures and the one or more information views (see paragraphs [0050-0051, 0056-0063, 0085-0089]). The tool (100) also includes an executor engine component (140) that uses the plurality of definitions created by the builder component (130) to assemble the information system at run time (see paragraphs [0033, 0064 and 0131]). Also, after the information system is assembled, the information system is modifiable or expandable by one or more additional transaction structures and/or one or more additional information views without any down time (See paragraphs [0003, 0123-0125]). See also original claims 1-5; the Abstract; and Figures 1, 3 and 4.

Claim 17:

With regard to claim 17, a method of building an information system onto a computer-readable medium is provided (see paragraphs [0001] and [0004]). The method recites creating a plurality of definitions with a builder (130) that uses one or more transaction structures and one or more information views that form a business process (see paragraphs [0050-0051, 0056-0063, 0085-0089]). The method also requires assembling the information system with an executor engine (140) that uses the plurality of definitions created by the builder to assemble the information system at run time (see paragraphs [0033, 0064 and 0131]). Claim 17 further recites modifying and expanding the information system without any downtime, if one or more additional transaction structures and/or one or more additional information views are received by the builder (See paragraphs [0003, 0123-0125]). See also original claims 1-5; the Abstract; and Figures 1, 3 and 4.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether claims 10-16 are directed to non-statutory subject matter and more particularly:

1. Whether claim 10 is directed to non-statutory subject matter.

B. Whether Turner et al. (U.S. Patent No. 6,230,309) anticipates claims 10-14, 16 and 17 and more particularly:

1. Whether Turner et al. discloses all of the features of claim 10.

2. Whether Turner et al. discloses all of the features of claim 17.

VII. ARGUMENT

A. Claims 10-16 are directed to statutory subject matter under 35 U.S.C. §101.

1. Claim 10 is directed to statutory subject matter.

Claim 10 is directed to a computer-readable medium having stored thereon a tool for building an information system. As noted in the January 29, 2009 Amendment, a telephonic interview took place between the Examiner, Ms. Anna Cheng Deng, and Applicants' representatives James Larson and Amol Kavathekar on January 23, 2009. In the interview, Applicants' representatives proposed amending the preamble of claim 10 to recite "A computer-readable medium having stored thereon a tool for building an information system". The Examiner agreed that this would overcome the rejection if the specification made no reference to varying signals or waves. Also, in the February 9, 2009 Advisory Action, the Examiner entered the above amendments to claim 10 but did not comment on the amendments in the Continuation Sheet of the Advisory Action. Accordingly, Applicants assert that the amendments to claim 10 overcome the statutory subject matter rejection and were found acceptable by the Examiner. Reversal of the rejection is respectfully requested for at least the foregoing reasons.

B. Claims 10-14, 16 and 17 are not anticipated under 35 U.S.C. §102(e) in view of Turner et al. (U.S. Patent No. 6,230,309).

Claims 10-14, 16 and 17 were rejected under 35 U.S.C. §102(e) as being anticipated by Turner et al. Appellants respectfully request reversal of the rejection for at least the following reasons.

1. Turner et al. fails to disclose all of the features of claim 10.

In particular, nowhere does Turner et al. disclose or suggest that after the information system is assembled, the information system is modifiable or expandable by one or more additional transaction structures and/or one or more additional information views without any down time.

Both the September 29, 2008 Office Action and the February 9, 2009 Advisory Action rely on column 15, lines 12-38 and Figure 7B of Turner et al. as disclosing that after the information system is assembled, the information system is modifiable or expandable by one or more additional transaction structures and/or one or more additional information views without

any down time (See page 5 of the September 29, 2008 Office Action and the Continuation Sheet of the February 9, 2009 Advisory Action).

However, column 15, lines 12-38 and Figure 7B of Turner et al. are related to inputted data values that can not be considered as the transaction structures and information views of claim 10. As recited in claim 10, transaction structures and information views form a business process and are used by the builder component to create a plurality of definitions which are then used to assemble the information system at runtime. Also, transaction structures are defined in the present application as the building blocks of an information system and information views are defined as governing the information presentation function of the information system (see paragraphs [0057] and [0086] of the present application). Accordingly, one skilled in the art would know that transaction structures and information view are used to define the infrastructure/framework of an information system and are not inputted data values that are managed by specific component objects within a created infrastructure/framework of an information system.

In contrast, column 15, lines 12-38 of Turner et al. discloses a Detail Operation Effect 68 that causes information about a current row in an Application View 80 to be refreshed or expanded by input values and an Update Operation Effect 68 that causes values in a current view in the Application View 80 to be updated by input values. Thus, this portion of Turner et al. merely describes how data values inputted after the application is assembled are managed by specific component objects of the application that are arranged within an assembled application infrastructure, and does not suggest that the data values inputted after the application is built are used to modify or expand the application infrastructure. The ability to refresh, add or modify rows within the framework of an already built application infrastructure does not equate to transaction structures or information views that define the framework of an information system.

Moreover, these inputted data values have no relation to the user declarations described in column 4, lines 41-67 of Turner et al. that are also interpreted by the Examiner as the transaction structures and the information views in rejecting other features of claim 10. Whereas, the inputted data values described in column 15, lines 12-38 of Turner et al. are used to refresh, add or modify rows within the framework of an already built application infrastructure, the user declarations are used to actually build an application infrastructure. Thus, column 15,

lines 12-38 of Turner et al. does not disclose the use of transaction structures or information views. Accordingly, nowhere does this portion of Turner et al. disclose or suggest that after an information system is assembled, the information system is modifiable or expandable by one or more additional transaction structures and/or one or more information views without any downtime, as required by claim 10.

Figure 7B of Turner et al. is a portion of a schematic block diagram of a general example of an Application Design model that shows an Application View Design area 42 that covers the entity types used to hold the definition of Application Views, a Component Description 44 that covers the features of components exploited by Application Views, and a Runtime Data Management area 46 that describes the objects which hold actual instances of Application Views. Nowhere does Figure 7B or the other portions of the Application Design model shown in Figures 7A and 7C of Turner et al. disclose or suggest that after the information system is assembled, the information system is modifiable or expandable by one or more additional transaction structures and/or one or more additional information views (which are used to create a plurality of definitions for assembling an information system) without any down time.

Also, column 4, lines 41-67 and column 5, lines 20-37 of Turner et al., cited by the Examiner as teaching the builder component and the executor engine component of claim 10, does not disclose or suggest that after the information system is assembled, the information system is modifiable or expandable by one or more additional transaction structures and/or one or more additional information views without any down time.

In particular, column 4, lines 41-67 of Turner et al. discloses a design tool that uses inputted user declarations to assemble component objects to form an object-based computer system application. The design tool includes a declarative user input interface mechanism and a design engine. The declarative user input interface mechanism allows a user to input user declarations that specify operative interaction between component objects. The design engine, in response to input user declarations, generates an application design definition that models an application infrastructure for managing component object interactions (see column 4, lines 41-51 of Turner et al.). Thus, the design tool of Turner et al. allows a user to input user declarations for building an application infrastructure made up of component objects.

Column 5, lines 20-37 of Turner et al. discloses a runtime tool configured to interpret the application design definition to generate application view instances for managing runtime

component object interactions. The runtime tool includes an application engine that is configured to create application view instances from respective application view definitions for managing runtime component object interactions for the application (see column 5, lines 26-43). Thus, the runtime tool of Turner et al. allows a user, once the application infrastructure is built, to input data values to manage the interaction between component objects.

However, nowhere does column 4, lines 41-67, column 5, lines 20-37 or any other portion of Turner et al. disclose that after the application infrastructure is assembled, the application infrastructure is modifiable or expandable by one or more user declarations without any down time. Thus, nowhere does Turner et al. disclose or suggest that after the information system is assembled, the information system is modifiable or expandable by one or more additional transaction structures and/or one or more additional information views without any down time.

For at least the foregoing reasons, Turner et al. does not disclose the features required by claim 10, and would not enjoy any of the advantages provided by the claimed invention. Appellants respectfully assert that claim 10 and its dependents are allowable over the Turner et al. reference. Reversal of the rejection is respectfully requested for at least the foregoing reasons.

2. Turner et al. fails to disclose all of the features of claim 17.

In particular, nowhere does Turner et al. disclose or suggest modifying and expanding the information system without any downtime, if one or more additional transaction structures and/or one or more additional information views are received by the builder.

Both the September 29, 2008 Office Action and the February 9, 2009 Advisory Action rely on column 15, lines 12-38 and Figure 7B of Turner et al. as disclosing modifying and expanding the information system without any downtime, if one or more additional transaction structures and/or one or more additional information views are received by the builder (See page 8 of the September 29, 2008 Office Action and the Continuation Sheet of the February 9, 2009 Advisory Action).

However, column 15, lines 12-38 and Figure 7B of Turner et al. are related to inputted data values that can not be considered as the transaction structures and information views of claim 17. As recited in claim 17, transaction structures and information views form a business

process and are used by a builder to create a plurality of definitions which are then used to assemble the information system at runtime. Also, transaction structures are defined in the present application as the building blocks of an information system and information views are defined as governing the information presentation function of the information system (see paragraphs [0057] and [0086] of the present application). Accordingly, one skilled in the art would know that transaction structures and information view are used to define the infrastructure/framework of an information system and are not inputted data values that are managed by specific component objects within a created infrastructure/framework of an information system.

In contrast, column 15, lines 12-38 of Turner et al. discloses a Detail Operation Effect 68 that causes information about a current row in an Application View 80 to be refreshed or expanded by input values and an Update Operation Effect 68 that causes values in a current view in the Application View 80 to be updated by input values. Thus, this portion of Turner et al. merely describes how data values inputted after the application is assembled are managed by specific component objects of the application that are arranged within an assembled application infrastructure, and does not suggest that the data values inputted after the application is built are used to modify or expand the application infrastructure. The ability to refresh, add or modify rows within the framework of an already built application infrastructure does not equate to transaction structures or information views that define the framework of an information system.

Moreover, these inputted data values have no relation to the user declarations described in column 4, lines 41-67 of Turner et al. that are also interpreted by the Examiner as the transaction structures and the information views in rejecting other features of claim 10. Whereas, the inputted data values described in column 15, lines 12-38 of Turner et al. are used to refresh, add or modify rows within the framework of an already built application infrastructure, the user declarations are used to actually build an application infrastructure. Thus, column 15, lines 12-38 of Turner et al. does not disclose the use of transaction structures or information views. Accordingly, nowhere does this portion of Turner et al. disclose or suggest modifying and expanding the information system without any downtime, if one or more additional transaction structures and/or one or more additional information views are received by the builder, as required by claim 17.

Figure 7B of Turner et al. is a portion of a schematic block diagram of a general example of an Application Design model that shows an Application View Design area 42 that covers the entity types used to hold the definition of Application Views, a Component Description 44 that covers the features of components exploited by Application Views, and a Runtime Data Management area 46 that describes the objects which hold actual instances of Application Views. Nowhere does Figure 7B or the other portions of the Application Design model shown in Figures 7A and 7C of Turner et al. disclose or suggest modifying and expanding the information system without any downtime, if one or more additional transaction structures and/or one or more additional information views (which are used to create a plurality of definitions for assembling an information system) are received by the builder.

Also, column 4, lines 41-67 and column 5, lines 20-37 of Turner et al., cited by the Examiner as teaching the other features of claim 17, does not disclose or suggest modifying and expanding the information system without any downtime, if one or more additional transaction structures and/or one or more additional information views are received by the builder.

In particular, column 4, lines 41-67 of Turner et al. discloses a design tool that uses inputted user declarations to assemble component objects to form an object-based computer system application. The design tool includes a declarative user input interface mechanism and a design engine. The declarative user input interface mechanism allows a user to input user declarations that specify operative interaction between component objects. The design engine, in response to input user declarations, generates an application design definition that models an application infrastructure for managing component object interactions (see column 4, lines 41-51 of Turner et al.). Thus, the design tool of Turner et al. allows a user to input user declarations for building an application infrastructure made up of component objects.

Column 5, lines 20-37 of Turner et al. discloses a runtime tool configured to interpret the application design definition to generate application view instances for managing runtime component object interactions. The runtime tool includes an application engine that is configured to create application view instances from respective application view definitions for managing runtime component object interactions for the application (see column 5, lines 26-43). Thus, the runtime tool of Turner et al. allows a user, once the application infrastructure is built, to input data values to manage the interaction between component objects.

However, nowhere does column 4, lines 41-67, column 5, lines 20-37 or any other portion of Turner et al. disclose modifying and expanding the application infrastructure without any downtime, if one or more additional user declarations are received by the design tool. Thus, nowhere does Turner et al. disclose or suggest modifying and expanding the information system without any downtime, if one or more additional transaction structures and/or one or more additional information views are received by the builder.

For at least the foregoing reasons, Turner et al. does not disclose the features required by claim 17, and would not enjoy any of the advantages provided by the claimed invention. Appellants respectfully assert that claim 17 and its dependents are allowable over the Turner et al. reference. Reversal of the rejection is respectfully requested for at least the foregoing reasons.

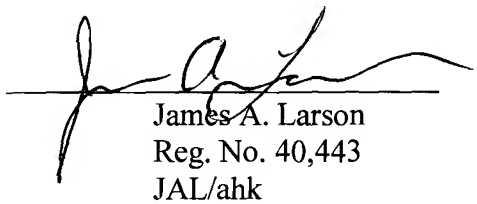
CONCLUSION

Appellants submit that the rejections are untenable for the reasons set forth above and should be reversed.

Respectfully submitted,

HAMRE, SCHUMANN, MUELLER
& LARSON P.C.
P.O. Box 2902
Minneapolis, MN 55402-0902
612.455.3800

Dated: May 26, 2009

By 

James A. Larson
Reg. No. 40,443
JAL/ahk

52835
Customer Number

VIII. CLAIMS APPENDIX

1-9. (Cancelled)

10. (Previously Presented) A computer-readable medium having stored thereon a tool for building an information system, the tool comprising:

a builder component that receives one or more transaction structures and one or more information views that form a business process, and creates a plurality of definitions using the one or more transaction structures and the one or more information views; and

an executor engine component that uses the plurality of definitions created by the builder component to assemble the information system at run time;

wherein after the information system is assembled, the information system is modifiable or expandable by one or more additional transaction structures and/or one or more additional information views without any down time.

11. (Previously Presented) The tool of claim 10, wherein each of the one or more transaction structures comprises one or more data containers, one or more input fields, one or more graphical user interface definitions, one or more validation statements, one or more process maps, or one or more print formats.

12. (Previously Presented) The tool of claim 10, further comprising a database wherein the definitions created by the builder component are stored as definition data in the database and accessed by the executor engine component.

13. (Previously Presented) The tool of claim 10, wherein after the information system is assembled, the builder component is capable of receiving the one or more additional transaction structures and/or the one or more additional information views and creating one or more additional definitions, and the executor engine component is capable of assembling the one or more additional definitions created by the builder component to modify and expand the existing information system without any down time.

14. (Previously Presented) The tool of claim 10, wherein the executor engine component comprises:

a process request server that processes one or more transaction or information requests; and

a graphical user interface layer that presents a user interface of the information system to a user, receives one or more transaction or information requests, and submits the one or more transaction or information requests to the process request server.

15. (Previously Presented) The tool of claim 10, wherein the information system is domain-neutral.

16. (Previously Presented) The tool of claim 10, wherein the one or more transaction structures, the one or more information views, the one or more additional transaction structures or the one or more additional information views is received via the Internet.

17. (Previously Presented) A method of building an information system onto a computer-readable medium comprising:

creating a plurality of definitions with a builder that uses one or more transaction structures and one or more information views that form a business process;

assembling the information system with an executor engine that uses the plurality of definitions created by the builder to assemble the information system at run time;

modifying and expanding the information system without any downtime, if one or more additional transaction structures and/or one or more additional information views are received by the builder.

IX. EVIDENCE APPENDIX

Not Applicable.

X. RELATED PROCEEDINGS APPENDIX

None